INVENTION TITLE

System and Method to distribute reasoning and pattern matching in forward and backward chaining rule engines

DESCRIPTION

Heading

Summary

[Para 1] The invention solves performance issues previously ignored by researchers in the field of artificial intelligence. Using existing techniques, a rule engine implementing Dr. Forgy's original RETE algorithm cannot reason over extremely large datasets or partitioned data. The invention solves the following limitations.

- Dividing the pattern matching between multiple rule engines and sharing the partial and complete matches efficiently.
- Efficiently dividing the pattern matching across several rule engines dynamically.
- Efficiently distributing the indexes of the working memory across multiple machines.
- Efficiently distributing one or more conditional elements of a rule to remote systems and dividing the work across a cluster of computer systems.

[Para 2] In generally, one form of the invention is a computer program which implements the RETE extension to perform distributed reasoning. The device would receive input from an external source and derive a conclusion based a set of rules. The rule defines concrete actions, which should be executed when all the conditions are satisfied. The device can be used in a variety of applications like order management, regulatory compliance, military command control applications or business process automation. The invention is particularly well suited to large real-time systems like military command control systems or security trading systems, which partition data across several systems in multiple physical locations.

Detailed Description of the preferred embodiment

[Para 3] Distributed reasoning, unlike collaborative agents provides methods by which multiple rule engines reason over large datasets in real-time. Whereas collaborative agents use agents to reason over discrete problems, it cannot reason over large sets of data. Furthermore, collaborative agents" techniques require the rule engine to load all necessary data within the same engine. For small datasets, these techniques prove to be powerful, but they do not scale for large datasets.

[Para 4] A forward chaining rule engine utilizing RETE algorithm, can reason over large datasets when nodes are distributed to other systems. This allows the system to match on a specific instance of a class (also called business object) without requiring the object instance reside in the same working memory and be locally available. It is important to note the engine will reason over shared data resident in other engines using data steams or indexes. This reduces the memory requirements and improves efficiency. It also enables the engines to share all or part of their reasoning network and distribute the process. Distributing nodes of a set of rules across multiple systems allows each engine to perform pattern matching on a single object instance and route the results to the originating system. Unlike load balancing techniques, a true distributed reasoning system does not require all systems of a cluster to deploy identical sets of rules, which creates redundant rules within the environment and increases maintenance costs. In a distributed reasoning/distributed pattern matching system, each system deploys a different set of rules (also called module or rule set), based on its own needs and configuration requirements. At runtime, the rule engines monitor resource utilization and distribute nodes dynamically and on demand.

[Para 5] Prior techniques relied on load balancing techniques to prioritize and categorize atomic processes. This approach requires every system to have all required data locally, leading to multiple data caches. In a production environment with large datasets, each